Parametric families for Monte Carlo on binary spaces

Christian Schäfer[1,2]

November 15, 2011

In the context of adaptive Monte Carlo algorithms, we cannot directly generate independent samples from the distribution of interest but use a proxy which we need to be close to the target. Generally, such a proxy distribution is a parametric family on the sampling spaces of the target distribution. For continuous sampling problems in high dimensions, we often use the multivariate normal distribution as a proxy for we can easily parametrize it by its moments and quickly sample from it. The objective is to construct similarly flexible parametric families on binary sampling spaces too large for exhaustive enumeration.

Keywords Binary parametric families · Sampling correlated binary data

1 Introduction

1.1 Parametric families for Monte Carlo

We discuss parametric families on binary spaces against the backdrop of Monte Carlo applications. The construction of binary parametric families q_θ that can model and reproduce the dependence structure of the target distribution π is a difficult task, and many concepts of modeling multivariate binary data fail to provide parametric families that are suitable for adaptive Monte Carlo algorithms. Therefore, we do not only discuss workable families but also approaches that are impractical in order to provide a thorough review of all available methods.

1.2 Notation

We denote scalars in italic type, vectors in italic bold type and matrices in straight bold type. We write diag $[\boldsymbol{a}]$ for the diagonal matrix of the vector $\boldsymbol{a}$ and diag $[\mathbf{A}]$ for the main diagonal of the matrix $\mathbf{A}$. The determinant is denoted by det $[\mathbf{A}]$. We write $a_{i\bullet}$ and $a_{\bullet j}$ for the ith row and jth column of $\mathbf{A}$, respectively. We write $\mathbf{A} \succ 0$ to indicate that $\mathbf{A}$ is positive definite. Given as set M, we write $|M|$ for the number of its elements, $\overline{M}$ for its closure and $\mathbb{1}_M$ for its indicator function.

[1]Centre de Recherche en Économie et Statistique, 3 Avenue Pierre Larousse, 92240 Malakoff, France
[2]CEntre de REcherches en MAthématiques de la DEcision, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny 75775 Paris, France

We write $\mathbb{B} = \{0, 1\}$ for the binary space and denote by $d \in \mathbb{N}$ the generic dimension. Given a vector $\boldsymbol{\gamma} \in \mathbb{B}^d$ and an index set $I \subseteq \{1, \ldots, d\}$, we write $\boldsymbol{\gamma}_I \in \mathbb{B}^{|I|}$ for the sub-vector indexed by I and $\boldsymbol{\gamma}_{-I} \in \mathbb{B}^{d-|I|}$ for its complement. If I is a sequence $\{i, \ldots, j\}$ we use the more explicit notation $\boldsymbol{\gamma}_{i:j}$ instead of $\boldsymbol{\gamma}_I$ and γ_i if $I = \{i\}$.

We write $\boldsymbol{\gamma}_{I_1}$ and $\boldsymbol{\gamma}_{I_0}$ for $\boldsymbol{\gamma}$ with its components indexed by I set to $\mathbf{1}$ and $\mathbf{0}$, respectively. In particular, we frequently use the short notation $\boldsymbol{a}_{i\bullet}^\intercal \boldsymbol{\gamma}_{i_1}$ for $a_{ii} + \sum_{j=1}^{i-1} a_{ij}\gamma_j$ where $\mathbf{A}$ is a lower triangular matrix.

1.3 Data from the target distribution

In the sequel, let $d > 0$ denote the dimension of the binary space $\mathbb{B}^d = \{0, 1\}^d$. Adaptive Monte Carlo algorithms are generally able to produce a, not necessarily independent and possibly weighted, sample

$$\boldsymbol{w} = (w_1, \ldots, w_n) \in [0, 1]^n, \quad \mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\intercal \in \mathbb{B}^{n \times d}$$

from the target distribution π we want to emulate using a binary family. We define the index set $D = \{1, \ldots, d\}$ and denote by

$$\bar{x}_i := \textstyle\sum_{k=1}^n w_k x_{k,i}, \quad \bar{x}_{i,j} := \sum_{k=1}^n w_k x_{k,i} x_{k,j}, \quad i, j \in D \tag{1}$$

the weighted first and second sample moments. We further define by

$$r_{i,j} := \frac{\bar{x}_{i,j} - \bar{x}_i \bar{x}_j}{\sqrt{\bar{x}_i(1 - \bar{x}_i)\bar{x}_j(1 - \bar{x}_j)}}, \qquad i, j \in D. \tag{2}$$

the weighted sample correlation.

1.4 Suitable parametric families

We first frame some properties making a parametric family suitable as sampling distribution in adaptive Monte Carlo algorithms.

(a) For reasons of parsimony, we want to construct a family of distributions with at most $\dim(\theta) \leq d(d + 1)/2$ parameters.

(b) Given a sample $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\intercal$ from the target distribution π, we need to estimate θ^* such that the binary family q_{θ^*} is close to π.

(c) We need to generate samples $\mathbf{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)^\intercal$ from the family q_θ. We need the rows of $\mathbf{Y}$ to be independent.

(d) For some algorithms, we need to evaluate the probability $q_\theta(\boldsymbol{y})$. For instance, we need $q_\theta(\boldsymbol{y})$ to compute importance weights or acceptance ratios in the context of Importance Sampling or Markov chain Monte Carlo, respectively.

(e) Analogously to the multivariate normal, we need our calibrated binary family q_{θ^*} to reproduce the marginals and covariance structure of π.

2 Distributions on binary spaces

Before we embark on the discussion of binary families, we make some observations which hold true for every binary distribution. The notation and results introduced in this section will be used throughout the rest of this work. Here, we denote by π some generic distribution on $\mathbb{B}^d$

Moments We use the short notation,

$$u_I(\boldsymbol{\gamma}) := \prod_{i\in I} \gamma_i, \qquad I \subseteq D,$$

for the product of all components index by I with $\prod_{i\in\emptyset} = 1$. Since $u_I(\boldsymbol{\gamma}) = 1$ iff $\boldsymbol{\gamma}_i = 1$ for all $i \in I$, u_I is the indicator function for the unit vector $\mathbf{1}_{|I|}$. We can characterize every distribution on $\mathbb{B}^d$ by $2^d - 1$ full probabilities

$$p_I := \mathbb{P}_\pi\left(\boldsymbol{\gamma}_I = 1, \boldsymbol{\gamma}_{D\setminus I} = 0\right), \qquad I \subseteq D$$

or by $2^d - 1$ cross-moments, that is marginal probabilities,

$$m_I := \mathbb{E}_\pi\left(u_I(\boldsymbol{\gamma})\right) = \mathbb{P}_\pi\left(\boldsymbol{\gamma}_I = \mathbf{1}\right), \qquad I \subseteq D.$$

In the following, we assume that $m_i \in (0, 1)$ for all $i \in D$, since for $m_i \in \{0, 1\}$, the component $\gamma_i = m_i$ is constant and therefore not part of the sampling problem.

For the product of components normalized to have zero mean and unit variance, we write

$$v_I(\boldsymbol{\gamma}) := \prod_{i\in I} (\gamma_i - m_i)/\sqrt{m_i(1 - m_i)}, \qquad I \subseteq D.$$

Note that $\mathbb{E}_\pi\left(v_{i,j}\right)$ is the correlation between γ_i and γ_j. Therefore, we call

$$c_I := \mathbb{E}_\pi\left(v_I(\boldsymbol{\gamma})\right)$$

the correlation of order $|I|$.

Marginals We use the notation

$$\pi_I(\boldsymbol{\gamma}_I) = \sum_{\boldsymbol{\xi}\in\mathbb{B}^{d-|I|}} \pi(\boldsymbol{\gamma}_I, \boldsymbol{\xi}), \qquad I \subseteq D.$$

for the marginal distributions. Note the connection to the cross-moments

$$\pi_I(\mathbf{1}_{|I|}) = \sum_{\boldsymbol{\xi}\in\mathbb{B}^{d-|I|}} \pi(\mathbf{1}_{|I|}, \boldsymbol{\xi}) = \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} u_I(\boldsymbol{\gamma})\ \pi(\boldsymbol{\gamma}) = m_I. \tag{3}$$

Representations Let π be the mass function of a binary distribution and suppose there is a bijective mapping $\tau\colon \mathbb{R} \supseteq V \to \pi(\mathbb{B}^d)$. There are coefficients $a_I \in \mathbb{R}$ such that

$$\pi(\boldsymbol{\gamma}) = \tau\left[\sum_{I\subseteq D} a_I \prod_{i\in I} \gamma_i\right]. \tag{4}$$

Proof. Immediate from the representation of the Dirac delta function as a product,

$$\pi(\boldsymbol{\gamma}) = \tau\left[\sum_{I\subseteq D} \delta_{\kappa^I}(\boldsymbol{\gamma})\tau^{-1}(\pi(\kappa^I))\right], \quad \delta_{\kappa^I}(\boldsymbol{\gamma}) = \prod_{i\in I} \gamma_i \prod_{i\in\{1,\ldots,d\}\setminus I} (1 - \gamma_i),$$

where κ^I denotes the vector with $\kappa_i^I = \mathbb{1}_I(i)$ for all $i \in \{1, \ldots, d\}$. $\square$

Constraints The general constraints on binary data are

$$\left(\sum_{i\in I} m_i - |I| + 1\right) \vee 0 \leq m_I \leq \min\{m_K \mid K \subseteq I\}, \tag{5}$$

where the upper bound is the monotonicity of the measure, and the lower bound follows from

$$\begin{aligned} |I| - 1 &= \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} (|I| - 1)\pi(\boldsymbol{\gamma}) \\ &\geq \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \left(\sum_{i\in I} \gamma_i - u_I(\gamma)\right)\pi(\boldsymbol{\gamma}) \\ &= \sum_{i\in I} m_i - m_I. \end{aligned}$$

In fact, m_I is a $|I|$-dimensional copula with respect to the expectations m_i for $i \in I$, see Nelsen (2006, p.45), and the inequalities (5) correspond to the Fréchet-Hoeffding bounds.

Sampling For sampling from a binary distribution π, we apply the chain rule factorization

$$\begin{aligned} \pi(\boldsymbol{\gamma}) &= \pi_{\{1\}}(\gamma_1) \prod_{i=2}^{d} \pi_{\{1:i\}}(\gamma_i \mid \boldsymbol{\gamma}_{1:i-1}) \\ &= \pi_{\{1\}}(\gamma_1) \prod_{i=2}^{d} \pi_{\{1:i-1\}}(\boldsymbol{\gamma}_{1:i-1})/\pi_{\{1:i\}}(\boldsymbol{\gamma}_{1:i}), \end{aligned} \tag{6}$$

which permits to sample a random vector component-wise, conditioning on the entries we already generated. We do not even need to compute the full decomposition (6), but only the conditional probabilities $\pi_{\{1:i\}}(\gamma_i = 1 \mid \boldsymbol{\gamma}_{1:i-1})$ defined by

$$\frac{\pi_{\{1:i\}}(\boldsymbol{\gamma}_{1:i-1}, 1)}{\pi_{\{1:i\}}(\boldsymbol{\gamma}_{1:i-1}, 1) + \pi_{\{1:i\}}(\boldsymbol{\gamma}_{1:i-1}, 0)}. \tag{7}$$

The full probability $\pi(\boldsymbol{\gamma})$ is then computed as a by-product of Procedure 1.

Procedure 1 Sampling via chain rule factorization

$\boldsymbol{y} = (0, \ldots, 0),\ p \leftarrow 1$
for $i = 1\ldots, d$ **do**
  $r \leftarrow \pi_{\{1:i\}}(\gamma_i = 1 \mid \gamma_{1:i-1})$
  sample $u \sim \mathcal{U}_{[0,1]},\ y_i \leftarrow \mathbb{1}_{[0,r]}(u)$
  $p \leftarrow \begin{cases} p \cdot r & \textbf{if} \quad y_i = 1 \\ p \cdot (1-r) & \textbf{if} \quad y_i = 0 \end{cases}$
end for
return $\boldsymbol{y}$, p

3 The product family

The simplest non-trivial distributions on $\mathbb{B}^d$ are certainly those having independent components.

3.1 Definition

For a vector $\mathbf{m} \in (0,1)^d$ of marginal probabilities, we define the product family

$$\begin{aligned} q_{\boldsymbol{m}}^{\mathrm{Prod}}(\boldsymbol{\gamma}) &:= \prod_{i\in D} m_i^{\gamma}(1-m_i)^{1-\gamma} \\ &= \prod_{i\in D}(1-m_i)\exp\left(\sum_{i\in D}\ell(m_i)\right). \end{aligned} \tag{8}$$

The second representation using the logit function

$$\ell\colon (0,1) \to \mathbb{R}, \quad \ell(p) = \log p - \log(1-p) \tag{9}$$

is useful to identify the product family as special case of more complex families.

3.2 Properties

We check the requirement list from Section 1.4:

(a) The product family is parsimonious with $\dim(\theta) = d$.

(b) The maximum likelihood estimator $\boldsymbol{m}^*$ is the sample mean (1).

(c) We easily sample from $q_{\boldsymbol{m}}^{\mathrm{Prod}}$, since (6) holds trivially.

(d) We easily evaluate the probability of a product of independent components.

(e) The family $q_{\boldsymbol{m}}^{\mathrm{Prod}}$ does not reproduce dependencies we might observe in the data $\mathbf{X}$.

The last point is a weakness which makes this simple family impractical when adaptive Monte Carlo algorithms are applied to challenging sampling problems. The product family $q_{\boldsymbol{m}}^{\mathrm{Prod}}$ is might often fail to mimick the target distribution π sufficiently well. Therefore, the rest of this paper deals with ideas on how to sample binary vectors with a given dependence structure.

3.3 Beyond the product family

There are, to our knowledge, two main strategies to produce binary vectors with correlated components.

(1) We can construct a generalized linear family which permits computation of its marginal distributions. We apply the chain rule factorization (6) and write q_θ as

$$q_\theta(\boldsymbol{\gamma}) = q_\theta(\gamma_1)\prod_{i=2}^{d} q_\theta(\gamma_i \mid \gamma_{1:i-1}), \tag{10}$$

which allows us to sample vectors component-wise.

(2) We sample from a multivariate auxiliary distribution h_θ dichotomize the samples, that is map them into $\mathbb{B}^d$. We call

$$q_\theta(\boldsymbol{\gamma}) = \int_{\tau^{-1}(\boldsymbol{\gamma})} h_\theta(\boldsymbol{v})d\boldsymbol{v} \tag{11}$$

a copula family since we exploit the copula structure of the underlying distribution to build a new parametric family. However, we refrain from working with explicit uniform marginals which is not all necessary (Mikosch, 2006).

In the following, we first study a few generalized linear families and then review a some copula approaches.

4 The linear quadratic family

Taking τ the identity mapping in (4), we obtain a full linear representation

$$\pi(\boldsymbol{\gamma}) = \textstyle\sum_{I \subseteq D} a_I \; u_I(\boldsymbol{\gamma}).$$

However, we cannot give a useful interpretation of the coefficients a_I. Bahadur (1961) derived the following representation:

Proposition 4.1. *We can write any binary distribution as*

$$\pi(\boldsymbol{\gamma}) = q^{Prod}_{\boldsymbol{m}}(\boldsymbol{\gamma}) \left(\textstyle\sum_{I \subseteq D} v_I(\boldsymbol{\gamma}) \; c_I\right),$$

where $\boldsymbol{m} = (m_1, \ldots, m_d)$ are the marginal probabilities.

Proof. For convenience, we provide the proof of Bahadur in Appendix 10. □

This decomposition, first discovered by Lazarsfeld, is a special case of a more general interaction theory (Streitberg, 1990) and allows for a reasonable interpretation of the parameters. Indeed, we have a product family times a correction term $1+\sum_{I \in \mathcal{I}_k} v_I(\boldsymbol{\gamma}) \; c_I$ where the coefficients are higher order correlations.

4.1 Definition

We can try to construct a more parsimonious family by removing higher order interaction terms. For additive approaches, however, we face the problem that a truncated representations do not necessarily define probability distributions since they might not be non-negative.

Still, for a symmetric matrix $\mathbf{A}$, we define the $d(d+1)/2$ parameter family

$$q^{\mathrm{LinQu}}_{\mathbf{A},a_0}(\boldsymbol{\gamma}) = \mu(a_0 + \boldsymbol{\gamma}^\intercal \mathbf{A} \boldsymbol{\gamma}), \tag{12}$$

where $\mu > 0$ is a normalizing constant and we set $a_0 = -(\min_{\boldsymbol{\gamma} \in \mathbb{B}^d} \boldsymbol{\gamma}^\intercal \mathbf{A} \boldsymbol{\gamma} \wedge 0)$. Since a_0 is the solution of an NP hard quadratic unconstrained binary optimization problem, this definition is of little practical value.

4.2 Moments

In virtue of the linear structure, we can derive explicit expressions for the cross-moments and marginal distributions, explicit meaning that the complexity is polynomial in d. The proofs are basic but rather tedious, so we moved them to the appendix section.

Next, we give a general formula yielding all cross-moments, including the normalizing constant.

Proposition 4.2. *For a set of indices $I \subseteq D$, we can write the corresponding cross-moment as*

$$m_I = \frac{1}{2^{|I|}} + \frac{\sum_{i \in I} \left[2 \sum_{j \in D} a_{i,j} + \sum_{j \in I \setminus \{i\}}^{d} a_{i,j} \right]}{2^{|I|}(4a_0 + \mathbf{1}^\intercal \mathbf{A} \mathbf{1} + \mathrm{tr}\,[\mathbf{A}])}.$$

For a proof see Appendix 10

Corollary 4.3. *The normalizing constant is*

$$\mu = 2^{-d+2}\left(4a_0 + \mathbf{1}^\intercal \mathbf{A}\mathbf{1} + \operatorname{tr}[\mathbf{A}]\right)^{-1},$$

and the expected value is

$$\mathbb{E}_{q^{LinQu}_{\mathbf{A},a_0}}(\gamma_i) = \frac{1}{2} + \frac{\sum_{k=1}^d a_{i,k}}{4a_0 + \mathbf{1}^\intercal \mathbf{A}\mathbf{1} + \operatorname{tr}[\mathbf{A}]}.$$

The mean m_i is close to $1/2$ unless the row $\boldsymbol{a}_i$ dominates the matrix. Therefore, if $\mathbf{A}$ is non-negative definite, the marginal probabilities m_i can hardly take values at the extremes of the unit interval.

4.3 Marginals

For the marginal distributions

$$q^{(1:k)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_{1:k}) = \sum_{\boldsymbol{\xi}\in\mathbb{B}^{d-(k+1)}} q_{\mathbf{A},a_0}(\boldsymbol{\gamma}_{1:k}, \boldsymbol{\xi})$$

there are explicit and recursive formulas. Hence, we can compute the chain rule decomposition (6) which in turn allows to sample from the family.

Proposition 4.4. *For the marginal distribution holds*

$$q^{(1:k)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_{1:k}) = \mu 2^{d-k-2} s_k(\boldsymbol{\gamma}_{1:k}),$$

where

$$\begin{aligned} s_k(\boldsymbol{\gamma}_{1:k}) = 4a_0 &+ \sum_{i=1}^k \gamma_i \left(\sum_{j=1}^k \gamma_j a_{i,j} + \sum_{j=k+1}^d a_{i,j}\right) \\ &+ \sum_{i=k+1}^d \sum_{j=k+1}^d a_{i,j} + \sum_{i=k+1}^d a_{i,i}. \end{aligned}$$

For a proof see Appendix 10

Recall the connection between marginal distributions and moments we observed in (3). For $\boldsymbol{\gamma}_I = \mathbf{1}$ we obtain

$$\begin{aligned} s_I(\mathbf{1}_k) &= 4a_0 + 4\sum_{i\in I}(\sum_{j\in I} a_{i,j} + \sum_{j\in I^c} a_{i,j}) \\ &\quad + \sum_{i\in I^c}\sum_{j\in I^c} a_{i,j} + \sum_{i\in I^c} a_{i,i} \\ &= 4a_0 + \sum_{i\in D}\sum_{j\in D} a_{i,j} + \sum_{i\in D} a_{i,i} + 3\sum_{i\in I}\sum_{j\in I} a_{i,j} \\ &\quad + 2\sum_{i\in I}\sum_{j\in I^c} a_{i,j} - \sum_{i\in I} a_{i,i} \\ &= 4a_0 + \mathbf{1}^\intercal \mathbf{A}\mathbf{1} + \operatorname{tr}[\mathbf{A}] + \\ &\quad \sum_{i\in I}\left[2\sum_{j\in D} a_{i,j} + \sum_{j\in I\setminus\{i\}} a_{i,j}\right], \end{aligned}$$

and $\pi_I(\mathbf{1}_k) = \mu 2^{d-|I|-2} s_I(\mathbf{1}_k)$ is indeed the expression for the cross-moments in Proof of Proposition 4.2.

4.4 Fitting the parameter

Given a sample $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\intercal \sim \pi$ from the target distribution, we can determine a_0 and a matrix $\mathbf{A}$ such that the family $q^{\mathrm{LinQu}}_{\mathbf{A},a_0}$ fits the first and second sampling moments

$$\bar{x}_{\{i,j\}} = n^{-1} \textstyle\sum_{k=1}^{n} x_{k,i} x_{k,j}, \quad i, j \in D$$

by solving a linear system of dimension $d(d+1)/2 + 1$. We first use the bijection

$$\tau \colon D \times D \to \{1, \ldots, d(d+1)/2\}, \quad \tau(i,j) = i(i-1)/2 + j$$

to map symmetric matrices into $\mathbb{R}^{(d+1)\,d/2}$. Precisely, for the matrices $\mathbf{A}$ and $\overline{\mathbf{X}}$, we define the vectors

$$\hat{a}_{\tau(i,j)} := a_{i,j}, \quad \hat{x}_{\tau(i,j)} := \bar{x}_{i,j}$$

and the design matrix

$$\hat{s}_{\tau(i,j),\tau(k,l)} := 2^{\mathbb{1}_{\{i,j\}}(k) + \mathbb{1}_{\{i,j,k\}}(l)}.$$

Note that $|\hat{\boldsymbol{a}}| = \mathbf{1}^\intercal \mathbf{A} \mathbf{1} + \mathrm{tr}\,[\mathbf{A}]$. We then equate the distribution moments to the sample moments and normalize such that

$$2^{d-2}(\mathbf{I}\, a_0 + \frac{1}{4}\, \hat{\mathbf{S}} \hat{\boldsymbol{a}}) = \hat{\boldsymbol{x}}, \quad 2^{d-2}(4a_0 + |\hat{\boldsymbol{a}}|) = 1. \tag{13}$$

The solution of the linear system

$$\begin{pmatrix} \hat{\boldsymbol{a}}^* \\ a_0^* \end{pmatrix} = 2^{-d+2} \begin{bmatrix} \frac{1}{4}\,\hat{\mathbf{S}} & \mathbf{1} \\ 4\,\mathbf{1}^\intercal & 1 \end{bmatrix}^{-1} \begin{pmatrix} \hat{\boldsymbol{x}} \\ 1 \end{pmatrix}$$

is finally transformed back into a symmetric matrix $\mathbf{A}^*$. Since the design matrix does not depend on the data, fitting several parameters to different data on the same space $\mathbb{B}^d$ is extremely fast.

4.5 Properties

We check the requirement list from Section 1.4:

(a) The linear family is sufficiently parsimonious having dimension $\dim(\theta) = d(d+1)/2$.

(b) We can fit the parameters $\mathbf{A}$ and a_0 via method of moments. However, the fitted function $q^{\mathrm{LinQu}}_{\mathbf{A}^*,a_0^*}(\boldsymbol{\gamma})$ is usually not a distribution.

(c) We can sample via chain rule factorization.

(d) We can evaluate $q^{\mathrm{LinQu}}_{\mathbf{A},a_0}(\boldsymbol{y})$ via chain rule factorization while sampling.

(e) The family $q^{\mathrm{LinQu}}_{\mathbf{A},a_0}$ reproduces the mean and correlations of the data $\mathbf{X}$.

Since in applications, the fitted matrix $\mathbf{A}^*$ is hardly ever positive definite, we cannot use the linear family in an adaptive Monte Carlo context. As other authors (Park et al., 1996; Emrich and Piedmonte, 1991) remark, additive representations like Proposition 4.1 are instructive but we cannot derive practical families from them.

5 The exponential quadratic family

If $\pi(\boldsymbol{\gamma}) > 0$ for all $\boldsymbol{\gamma} \in \mathbb{B}^d$, we can use $\tau = \exp$ in (4) and obtain a full log-linear representation

$$\pi(\boldsymbol{\gamma}) = \exp\left(\sum_{I \subseteq D} a_I \, u_I(\boldsymbol{\gamma})\right).$$

Note that we assume the probability mass function π is assumed to be log-linear in the parameters a_I. In the context of contingency tables the term "log-linear family" refers to the assumption that the marginal probabilities m_I are log-linear in the higher order marginals.

Remark Contingency table analysis is a well studied approach to modeling discrete data (Bishop et al., 1975; Christensen, 1997). For binary data, the underlying sampling distribution is assumed to be multinomial which requires an enumeration of the state space we want to avoid. Gange (1995) uses the Iterative Proportional Fitting algorithm (Haberman, 1972) from log-linear interaction theory to construct a binary distribution with given marginal probabilities. The fitting procedures, however, require storage of all configurations $\pi_I(\boldsymbol{\gamma}_I)$ and the construction of the joint posterior from the fitted marginal probabilities. The method is powerful and exact but computationally infeasible even for moderate dimensions.

5.1 Definition

Removing higher order interaction terms, we can construct a $d(d+1)/2$ parameter family

$$q_{\mathbf{A}}^{\mathrm{ExpQu}}(\boldsymbol{\gamma}) := \mu \exp(\boldsymbol{\gamma}^\intercal \mathbf{A} \boldsymbol{\gamma}), \tag{14}$$

where $\mathbf{A}$ is a symmetric matrix and $\mu := [\sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \exp(\boldsymbol{\gamma}^\intercal \mathbf{A} \boldsymbol{\gamma})]^{-1}$. We recognize the product family (8) as the special case $\mu = \prod_{i \in D}(1 - m_i)^d$ and $\mathbf{A} = \operatorname{diag}[\ell(\boldsymbol{m})]$.

5.2 Marginals

The moments or marginal distributions of $q_{\mathbf{A}}$ are sums of exponentials which, in general, do not simplify to expressions that are polynomial in d. Therefore, we cannot perform a chain rule factorization (6) to sample from the family.

Cox and Wermuth (1994) proposed the following second degree Taylor approximations to the marginal distributions which are again of the form (14).

Proposition 5.1. *We write the parameter* $\mathbf{A}$ *as*

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}' & \boldsymbol{b}^\intercal \\ \boldsymbol{b} & c \end{pmatrix}, \tag{15}$$

and define the parameters

$$\begin{aligned} \tilde{\mathbf{A}}_{d-1} &= \mathbf{A}' + \left(1 + \tanh(\tfrac{c}{2})\right) \operatorname{diag}[\boldsymbol{b}] + \tfrac{1}{2}\operatorname{sech}^2(\tfrac{c}{2}) \boldsymbol{b}\boldsymbol{b}^\intercal, \\ \tilde{\mu}_{d-1} &= \mu(1 + \exp(c)) \end{aligned}$$

Then $q_{\tilde{\mathbf{A}}_{d-1}}(\boldsymbol{\gamma}_{1:d-1})$ *is the second degree Taylor approximation to the marginal distribution* $q_{\mathbf{A}_{1:d-1}}(\boldsymbol{\gamma}_{1:d-1})$*. For a proof see Appendix 10.*

If we recursively compute $q_{\tilde{\mathbf{A}}_{d-1}}, \ldots, q_{\tilde{\mathbf{A}}_1}$, we can derive approximate conditional probabilities using (7). Precisely, we have

$$q_{\tilde{\mathbf{A}}_i}(\gamma_i = 1 \mid \boldsymbol{\gamma}_{1:i-1}) = \ell^{-1}(\tilde{c}_i + \tilde{\boldsymbol{b}}_i^\intercal \gamma_{1:i-1}), \tag{16}$$

where $\ell^{-1}(x) = (1 + \exp(-x))^{-1}$ and $\tilde{c}_i$, $\tilde{\boldsymbol{b}}_i$ are parts of the matrix $\tilde{\mathbf{A}}_i$ according to the notation introduced in (15). In particular, (16) is a logistic regression. We come back to this class of families in the following Section 6. We can sample from the proxy

$$\tilde{q}_{\tilde{\mathbf{A}}}(\boldsymbol{\gamma}) := \prod_{i \in D} q_{\tilde{\mathbf{A}}_i}(\gamma_i \mid \boldsymbol{\gamma}_{1:i-1}) \approx q_{\mathbf{A}}^{\mathrm{ExpQu}}(\boldsymbol{\gamma}),$$

which is close to the original exponential quadratic family. The goodness of the approximation might be improved by judicious permutation of the components. The approximation error is hard to control, however, since we repeatedly apply the second degree approximation and propagate initial errors.

5.3 Fitting the parameter

As in section 4.4, we use the bijection

$$\tau \colon D \times D \to \{1, \ldots, d(d+1)/2\}, \quad \tau(i,j) = i(i-1)/2 + j$$

to map symmetric matrices into $\mathbb{R}^{(d+1)\,d/2}$. Precisely, for the matrices $\mathbf{A}$ and $\overline{\mathbf{X}}$, we define the vectors

$$\hat{a}_{\tau(i,j)} := a_{i,j}, \quad \hat{x}_{\tau(i,j)} := \bar{x}_{i,j}.$$

We let $y_k = \log \pi(\boldsymbol{x}_k)$ for $k = 1, \ldots, n$ and fit the family solving the least square problem

$$\min_{\hat{\boldsymbol{a}} \in \mathbb{R}^{(d+1)\,d/2}} \left\| \hat{\mathbf{X}} \hat{\boldsymbol{a}} - \boldsymbol{y} \right\|_2$$

which yields the parameters

$$a_{i,j}^* = [(\hat{\mathbf{X}}^\intercal \hat{\mathbf{X}})^{-1} \hat{\mathbf{X}}^\intercal \boldsymbol{y}]_{\tau(i,j)}.$$

Note that in most adaptive Monte Carlo algorithms that involve importance sampling or Markov transitions, the probabilities $\pi(\boldsymbol{x}_k)$ of the target distribution are already computed such that the fitting procedure is rather fast.

5.4 Properties

We check the requirement list from Section 1.4:

(a) The log-linear family is sufficiently parsimonious with $\dim(\theta) = d(d+1)/2$.

(b) We can fit the parameter $\mathbf{A}$ via minimum least squares.

(c) We can sample from an approximation $\tilde{q}_{\tilde{\mathbf{A}}}(\boldsymbol{\gamma}) \approx q_{\mathbf{A}}(\boldsymbol{\gamma})$ to the log-linear family. However, we cannot control the approximation error.

(d) We can evaluate $q_{\mathbf{A},a_0}(\boldsymbol{y})$ up to the normalization constant $\boldsymbol{\mu}$ which suffices for most adaptive Monte Carlo methods.

(e) The family $q_{\mathbf{A},a_0}$ reproduces the mean and correlations of the data $\mathbf{X}$.

6 The logistic conditionals family

In the previous section we saw that even for a rather simple non-linear family we cannot derive closed-form expressions for the marginal probabilities. Therefore, instead of computing the marginals for a d-dimensional family $q_\theta(\boldsymbol{\gamma})$, we directly fit univariate families

$$q_\theta(\gamma_i = 1 \mid \gamma_{1:i-1}), \qquad i \in D$$

to the conditional probabilities $\pi(\gamma_i = 1 \mid \gamma_{1:i-1})$ of the target function. Precisely, we postulate the logistic relation

$$\ell(\mathbb{P}_\pi(\gamma_i = 1)) = b_{i,i} + \sum_{j=1}^{i-1} b_{i,j}\gamma_j, \quad i \in D$$

for the marginal probabilities where ℓ is the logit function defined in (9).

6.1 Definition

For a d-dimensional lower triangular matrix $\mathbf{B}$, we define the logistic conditionals family as

$$\begin{aligned} q_{\mathbf{B}}^{\mathrm{LogCo}}(\boldsymbol{\gamma}) &:= \prod_{i\in D} q^{\mathrm{Prod}}_{p(b_{i,i}+\boldsymbol{b}_{i,1:i-1}^\intercal \boldsymbol{\gamma}_{1:i-1})}(\gamma_i) \qquad (17)\\ &= \exp\left[\sum_{i\in D}\left[\gamma_i(b_{i,i}+\boldsymbol{b}_{i,1:i-1}^\intercal\boldsymbol{\gamma}_{1:i-1}) - \log\Big(1+\exp(b_{i,i}+\boldsymbol{b}_{i,1:i-1}^\intercal\boldsymbol{\gamma}_{1:i-1})\Big)\right]\right] \end{aligned}$$

where q_p^{Prod} is the Bernoulli distribution and

$$p(x) = \ell^{-1}(x) = (1+\exp(-x))^{-1}$$

the logistic function. We immediately identify the product family $q_{\boldsymbol{m}}^{\mathrm{Prod}}$ as the special case $\mathbf{B} = \operatorname{diag}[\ell(\boldsymbol{m})]$. The logistic conditionals family is not in the exponential family.

Note that there are $d!$ possible logistic families and we arbitrarily pick one while there should be a permutation $\sigma(D)$ of the components which is optimal in a sense of nearness to the data. In practice, however, changing the parametrization does not seem to have a noticeably impact on the quality of the adaptive Monte Carlo algorithm.

6.2 Sparse logistic regressions

The major drawback of all multiplicative families is the fact that they do not have closed-form likelihood-maximizers such that the parameter estimation requires costly iterative fitting procedures. Therefore, we construct a sparse version of the logistic regression family which we can estimate faster than the saturated family.

Instead of fitting the parameter of the saturated family $q_{\boldsymbol{b}}^{\mathrm{LogCo}}(\gamma_i \mid \gamma_{1:i-1})$, we preferably work with a more parsimonious regression family like $q_{\boldsymbol{b}}^{\mathrm{LogCo}}(\gamma_i \mid \gamma_{L_i})$ for some index set $L_i \subseteq \{1,\dots,i-1\}$, where the number of predictors $\#L_i$ is typically smaller than $i-1$.

We solve this nested variable selection problem using some simple, fast to compute criterion. For ε about $\frac{1}{100}$, we define the index set

$$I := \{i = 1,\dots,d \mid \ \bar{x}_i \notin (\varepsilon, 1-\varepsilon)\}.$$

which identifies the components which have, according to the data, a marginal probability close to either boundary of the unit interval.

We do not fit a logistic regression for the components $i \in I$. We rather set $L_i = \emptyset$ and draw them independently, that is we set $b_{i,i} = \ell(\bar{x}_i)$ and $\boldsymbol{b}_{i,-i} = \mathbf{0}$ which corresponds to logistic conditionals family without predictors. The reason is twofold. Firstly, interactions do not really matter if the marginal probability is excessively small or large. Secondly, these components are prone to cause complete separation in the data or might even be constant.

For the conditional distribution of the remaining components $I^c = D \setminus I$, we construct parsimonious logistic regressions. For δ about $\frac{1}{10}$, we define the predictor sets

$$L_i := \{j = 1, \ldots, i-1 \mid \delta < |r_{i,j}|\}, \quad i \in I^c,$$

which identifies the components with index smaller than i and significant mutual association.

6.3 Fitting the parameter

Given a sample $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\intercal \sim \pi$ from the target distribution we regress $\boldsymbol{y}^{(i)} = \mathbf{X}_i$ on the columns $\mathbf{Z}^{(i)} = (\mathbf{X}_{1:i-1}, \mathbf{1})$, where the column $\boldsymbol{Z}_i^{(i)}$ yields the intercept to complete the logistic conditionals family.

We maximize the log-likelihood function $\ell(\boldsymbol{b}) = \ell(\boldsymbol{b} \mid \boldsymbol{y}, \mathbf{Z})$ of a weighted logistic regression family by solving the first order condition $\partial\ell/\partial\boldsymbol{\beta} = \mathbf{0}$. We find a numerical solution via Newton-Raphson iterations

$$-\frac{\partial^2 \ell(\boldsymbol{b}^{(r)})}{\partial \boldsymbol{b}\boldsymbol{b}^\intercal}(\boldsymbol{b}^{(r+1)} - \boldsymbol{b}^{(r)}) = \frac{\partial \ell(\boldsymbol{b}^{(r)})}{\partial \boldsymbol{b}}, \quad r > 0, \tag{18}$$

starting at some $\boldsymbol{b}^{(0)}$; see Procedure 2 for the exact terms. Other updating formulas like Iteratively Reweighted Least Squares or quasi-Newton iterations should work as well.

Procedure 2 Fitting the weighted logistic regressions

Input: $\boldsymbol{w} = (w_1, \ldots, w_n),\ \mathbf{X} = (x_1, \ldots, x_n)^\intercal,\ \mathbf{B} \in \mathbb{R}^{d \times d}$

for $i \in I^c$ **do**

  $\mathbf{Z} \leftarrow (\mathbf{X}_{L_i}, \mathbf{1}),\ \boldsymbol{y} \leftarrow \mathbf{X}_i,\ \boldsymbol{b}^{(0)} \leftarrow \mathbf{B}_{i,L_i \cup \{i\}}$

  repeat

    $p_k \leftarrow \ell^{-1}(\mathbf{Z}_k \boldsymbol{b}^{(r-1)})$ **for all** $k = 1, \ldots, n$

    $q_k \leftarrow p_k(1 - p_k)$ **for all** $k = 1, \ldots, n$

    $\boldsymbol{b}^{(r)} \leftarrow (\mathbf{Z}^\intercal \mathrm{diag}\,[\boldsymbol{w}]\,\mathrm{diag}\,[\boldsymbol{q}]\,\mathbf{Z} + \varepsilon \mathbf{I}_n)^{-1} \times$
    $(\mathbf{Z}^\intercal \mathrm{diag}\,[\boldsymbol{w}]) \left(\mathrm{diag}\,[\boldsymbol{q}]\,\mathbf{Z}\,\boldsymbol{b}^{(r-1)} + (\boldsymbol{y} - \boldsymbol{p})\right)$

  until $|b_j^{(r)} - b_j^{(r-1)}| < 10^{-3}$ for all j

  $\mathbf{B}_{i,L_i \cup \{i\}} \leftarrow \boldsymbol{b}$

end for

return $\mathbf{B}$

Sometimes, the Newton-Raphson iterations do not converge because the likelihood function is monotone and thus has no finite maximizer. This problem is caused by data with complete or quasi-complete separation in the sample points (Albert and Anderson, 1984). There are several ways to handle this issue.

(a) We just halt the algorithm after a fixed number of iterations and ignore the lack of convergence. Such proceeding, however, might cause uncontrolled numerical problems.

(b) Firth (1993) recommends the Jeffreys prior for its bias reduction but this option is computationally rather expensive. We might instead use a Gaussian prior with variance $1/\varepsilon > 0$ which adds a quadratic penalty term $\varepsilon \boldsymbol{b}^\intercal \boldsymbol{b}$ to the log-likelihood to ensure the target-function is convex.

(c) As we notice that some terms of $\boldsymbol{b}_i$ are growing beyond a certain threshold, we move the component i from the set of components with associated logistic regression family I^c to the set of independent components I.

In practice, we recommend to combine the approaches (c) and (d). In Procedure 2, we did not elaborate how to handle non-convergence, but added a penalty term to the log-likelihood, which causes the extra $\varepsilon \mathbf{I}_n$ in the Newton-Raphson update. Since we solve the update equation via Cholesky factorizations, adding a small term on the diagonal ensures that the matrix is indeed numerically decomposable.

6.4 Properties

We check the requirement list from Section 1.4:

(a) The logistic regression family is sufficiently parsimonious with $\dim(\theta) = d(d+1)/2$.

(b) We can fit the parameters $\boldsymbol{b}_i$ via likelihood maximization for all $i \in D$. The fitting is computationally intensive but feasible.

(c) We can sample $\boldsymbol{y} \sim q_{\mathbf{B}}^{\mathrm{LogCo}}$ via chain rule factorization.

(d) We can exactly evaluate $q_{\mathbf{B}}^{\mathrm{LogCo}}(\boldsymbol{y})$.

(e) The family $q_{\mathbf{B}}^{\mathrm{LogCo}}$ reproduces the dependency structure of the data $\mathbf{X}$ although we cannot explicitly compute the marginal probabilities.

7 The Gaussian copula family

In the preceding sections, we discussed three approaches based on generalized linear families. Now we turn to the second class of families we call copula families.

Let h_θ be a family of auxiliary distributions on $\mathcal{X}$ and $\tau\colon \mathcal{X} \to \mathbb{B}^d$ a mapping into the binary state space. We can sample from the copula family

$$q_\theta^{h,\tau}(\boldsymbol{\gamma}) = \int_{\tau^{-1}(\boldsymbol{\gamma})} h_\theta(\boldsymbol{v})\, d\boldsymbol{v}$$

by setting $\boldsymbol{y} = h(\boldsymbol{v})$ for a draw $\boldsymbol{v} \sim h_\theta$ from the auxiliary distribution.

7.1 Definition

Apparently, non-normal parametric distributions s_θ with at most $d(d-1)/2$ dependence parameters either have a very limited dependence structure or rather unfavorable properties (Joe, 1996). Therefore, the multivariate Gaussian distribution with

$$h_{\boldsymbol{\Sigma}}(\boldsymbol{v}) = (2\pi)^{-\frac{d}{2}} \, |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp(-\frac{1}{2}\, \boldsymbol{v}^\intercal \boldsymbol{\Sigma}^{-1} \boldsymbol{v}),$$

and mapping $\tau\colon \mathbb{R}^d \to \mathbb{B}^d$

$$\tau_{\boldsymbol{\mu}}(\boldsymbol{v}) = (\mathbb{1}_{(\infty,\mu_i]}(v_1), \ldots, \mathbb{1}_{(\infty,\mu_d]}(v_d)),$$

appears to be the natural and almost the only option for h_θ. The Gaussian copula family, denoted by $q^{\text{GauC}}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$, has already been discussed repeatedly in the literature (Emrich and Piedmonte, 1991; Leisch et al., 1998; Cox and Wermuth, 2002).

7.2 Moments

For $I \subseteq D$, the cross-moment or marginal probabilities is

$$\begin{aligned} m_I &= \textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} q_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{1}_I, \boldsymbol{\gamma}_{D\setminus I}) = \int_{\bigcup_{\boldsymbol{\gamma}\in\mathbb{B}^d}\{\tau_{\boldsymbol{\mu}}^{-1}(\mathbf{1}_I,\boldsymbol{\gamma}_{D\setminus I})\}} h_{\boldsymbol{\Sigma}}(\boldsymbol{v})\, d\boldsymbol{v} \\ &= \textstyle\int_{\times_{i\in I}\{\tau_{\mu_i}^{-1}(1)\}} h_{\boldsymbol{\Sigma}}(\boldsymbol{v})\, d\boldsymbol{v} = \int_{\times_{i\in I}(-\infty,\mu_i]} h_{\boldsymbol{\Sigma}}(\boldsymbol{v})\, d\boldsymbol{v}, \end{aligned}$$

where we used (3) in the first line. Thus, the first and second moment of $q_{(\boldsymbol{\mu},\boldsymbol{\Sigma})}$ are

$$m_i = \Phi_1(\mu_i), \quad m_{i,j} = \Phi_2(\mu_i, \mu_j; \sigma_{i,j})$$

where $\Phi_1(v_i)$ and $\Phi_2(v_i, v_j; \sigma_{i,j})$ denote the cumulative distribution functions of the univariate and bivariate normal distributions with zero mean, unit variance and correlation coefficient $\sigma_{i,j} \in [-1, 1]$.

7.3 Sparse Gaussian copulas

We can speed up the parameter estimation and improve the condition of $\boldsymbol{\Sigma}$, if we work with a parsimonious Gaussian copula. We can apply the same criterion we already introduced for the sparse logistic regression family. For ε about $\frac{1}{100}$, we define the index set

$$I := \{i = 1, \ldots, d \mid \ \bar{x}_i \notin (\varepsilon, 1-\varepsilon)\}.$$

which identifies the components which have a marginal probability close to either boundary of the unit interval.

We do not fit a any correlation parameters for the components in I but set $\sigma_{i,j} = 0$ for all $j \in D \setminus \{i\}$. Firstly, the correlation does not really matter if the marginal probability is excessively small or large. Secondly, we fit the parameter $\boldsymbol{\Sigma}$ by separately adjusting the bivariate correlations $\sigma_{i,j}$, and components with high correlations and extreme marginal probability lower the chance that $\boldsymbol{\Sigma}$ is positive definite.

For the remaining components $I^c = D\setminus I$, we construct parsimonious Gaussian copula. For δ about $\frac{1}{10}$, we define the association set

$$A := \{\{i, j\} \in I^c \times I^c \mid \delta < |r_{i,j}|\,,\, i \neq j\}$$

which identifies the components with significant correlation. For $i, j \in D \times D \setminus L$ we also set $\sigma_{i,j} = 0$ to accelerate the estimation procedure.

7.4 Fitting the parameter

We fit the family $q_{\mathrm{GauC}(\boldsymbol{\mu},\boldsymbol{\Sigma})}$ to the data by adjusting $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to the sample moments. Precisely, we solve the equations

$$\varPhi_1(\mu_i) = \bar{x}_i, \qquad i \in D \tag{19}$$
$$\varPhi_2(\mu_i, \mu_j; \sigma_{i,j}) = \bar{x}_{i,j}, \qquad (i,j) \in A \tag{20}$$

with sample mean $\bar{x}_i$ and $\bar{x}_{i,j}$ as defined in (1). We easily solve (19) by setting

$$\mu_i = \Phi_1^{-1}(\bar{x}_i), \qquad i \in D.$$

The difficult task is computing a feasible correlation matrix from (20). Recall the standard result (Johnson et al., 2002, p.255)

$$\frac{\partial \Phi_2(y_1, y_2; \sigma)}{\partial \sigma} = h_\sigma(y_1, y_2), \tag{21}$$

where h_σ denotes the density of the bivariate normal distribution. We obtain the following Newton-Raphson iteration

$$\alpha_{r+1} = \alpha_r - \frac{\Phi_2(\mu_i, \mu_j; \alpha_r) - \bar{x}_{i,j}}{h_{\alpha_r}(\mu_i, \mu_j)}, \quad (i,j) \in A, \tag{22}$$

starting at some $\alpha_0 \in (-1, 1)$. We use a fast series approximation (Drezner and Wesolowsky, 1990; Divgi, 1979) to evaluate $\Phi_2(\mu_i, \mu_j; \alpha)$. These approximations are critical when α_r comes very close to either boundary of $[-1, 1]$. The Newton iteration might repeatedly fail when restarted at the corresponding boundary $r_0 \in \{-1, 1\}$. This is yet another reason why it is preferable to work with a sparse Gaussian copula. In any event, $\Phi_2(y_1, y_2; \sigma)$ is monotonic in σ since (21), and we can switch to bi-sectional search if necessary.

Procedure 3 Fitting the dependency matrix

Input: $\bar{x}_i,\ \bar{x}_{i,j}$ **for all** $i, j \in D$

$\mu_i = \varPhi_{-1}(\bar{x}_i)$ **for all** $i \in D$
$\boldsymbol{\Sigma} = \mathbf{I}_d$
for $(i,j) \in A$ **do**
  repeat
    $$\sigma_{i,j}^{(r+1)} \leftarrow \sigma_{i,j}^{(r)} - \frac{\Phi_2(\mu_i, \mu_j; \sigma_{i,j}^{(r)}) - \bar{x}_{i,j}}{h_{\sigma_{i,j}^{(r)}}(\mu_i, \mu_j)}$$
  until $|\sigma_{i,j}^{(r)} - \sigma_{i,j}^{(r-1)}| < 10^{-3}$
end for
if not $\boldsymbol{\Sigma} \succ 0$ **then** $\boldsymbol{\Sigma} \leftarrow (\boldsymbol{\Sigma} + |\lambda|\, \mathbf{I}_d)/(1 + |\lambda|)$
return $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$

A rather discouraging shortcoming of the Gaussian copula family is that locally fitted correlation matrices $\boldsymbol{\Sigma}$ might not be positive definite for $d \geq 3$. This is due to the fact that an elliptical copula, like the Gaussian, can only attain the bounds (5) for $d < 3$, but not for higher dimensions.

We propose two ideas to obtain an approximate, but feasible parameter:

(1) We replace $\mathbf{\Sigma}$ by $\mathbf{\Sigma}^* = (\mathbf{\Sigma} + |\lambda|\,\mathbf{I})/(1 + |\lambda|)$, where λ is the smallest eigenvalue of the dependency matrix $\mathbf{\Sigma}$. This approach evenly lowers the local correlations to a feasible level and is easy to implement on standard software. Alas, we make an effort to estimate $d(d-1)/2$ dependency parameters, and in the end we might not get more than an product family.

(2) We can compute the correlation matrix $\mathbf{\Sigma}^*$ which minimizes the distance $\|\mathbf{\Sigma}^* - \mathbf{\Sigma}\|_F$, where $\|\mathbf{A}\|_F^2 = \operatorname{tr}[\mathbf{A}\mathbf{A}^\intercal]$. In other words, we construct the projection of $\mathbf{\Sigma}$ into the set of correlation matrices. Higham (2002) proposes an Alternating Projections algorithm to solve nearest-correlation matrix problems. Yet, if $\mathbf{\Sigma}$ is rather far from the set of correlation matrices, computing the projection is expensive and, according to our experience, leads to troublesome distortions in the correlation structure.

7.5 Properties

We check the requirement list from Section 1.4:

(a) The Gaussian copula family is sufficiently parsimonious with $\dim(\theta) = d(d+1)/2$.

(b) We can fit the parameters $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ via method of moments. The parameter $\mathbf{\Sigma}$ is not always be positive definite which might require additional effort it feasible.

(c) We can sample $\boldsymbol{y} \sim q^{\mathrm{GauC}}_{(\boldsymbol{\mu},\mathbf{\Sigma})}$ using $\boldsymbol{y} = \tau_\mu(\boldsymbol{v})$ with $\boldsymbol{v} \sim h_{\mathbf{\Sigma}}$.

(d) We cannot evaluate $q^{\mathrm{GauC}}_{\mathbf{B}}(\boldsymbol{y})$ since this requires computing a high-dimensional integral expression.

(e) The family $q^{\mathrm{GauC}}_{(\boldsymbol{\mu},\mathbf{\Sigma})}$ reproduces the mean and correlation structure of the data $\mathbf{X}$.

Obviously, we cannot use the Gaussian copula family in the context of importance sampling or Markov chain Monte Carlo, since evaluation of $q^{\mathrm{GauC}}_{(\boldsymbol{\mu},\mathbf{\Sigma})}(\boldsymbol{y})$ is not feasible. This family might be useful, however, in other adaptive Monte Carlo algorithms, for instance the Cross-Entropy method (Rubinstein, 1997) for combinatorial optimization.

8 The Poisson reduction family

Let $N = \{1,\ldots,n\}$ denote another index set with $n \gg d$. Approaches to generating binary vectors that do not rely on the chain rule factorization (6) are usually based on combinations of independent random variables

$$\boldsymbol{v} = (v_1,\ldots,v_n) \sim \otimes_{k\in N} h_{\theta_k}.$$

We define index sets $\mathcal{M} = \{S_i \in N \mid i \in D\}$ and generate the entry y_i via

$$\tau_i \colon \mathcal{X}^{|S_i|} \to \{0,1\}, \quad \tau_i(\boldsymbol{v}) = f(\boldsymbol{v}_{S_i}), \quad i \in D.$$

In the context of Gaussian copulas, the auxiliary distributions $h_{\theta_k} = h_\theta$ are d independent standard normal variables. Park et al. (1996) propose the following family based on sums of independent Poisson variables.

8.1 Definition

We define a Poisson family $q^{\mathrm{Poi}}_{(\mathcal{S},\lambda)}$ with auxiliary distribution

$$h_\lambda(\boldsymbol{v}) = \prod_{k\in N}(\lambda_k^{v_k} e^{-\lambda_k})/v_k!$$

and mapping $\tau\colon \mathbb{N}_0^n \to \mathbb{B}^d$

$$\tau_{\mathcal{S}}(\boldsymbol{v}) = (\mathbb{1}_{\{0\}}(\textstyle\sum_{k\in S_1} v_k), \dots, \mathbb{1}_{\{0\}}(\sum_{k\in S_d} v_k)).$$

8.2 Moments

For an index set $I \in D$, the cross-moments or marginal probabilities are

$$m_I = \mathbb{P}\left(\forall i \in I\colon \textstyle\sum_{k\in S_i} v_k = 0\right) = \exp(-\sum_{k\in\cap_{i\in I} S_i} \lambda_k).$$

Therefore, fitting via method of moments is possible.

Proposition 8.1. *For $\boldsymbol{\gamma} \in \mathbb{B}^d$, define the index sets*

$$D_0 = \{i \in D \mid \gamma_i = 0\}, \quad D_1 = \{i \in D \mid \gamma_i = 1\},$$

and the families of subsets $\mathcal{I}_t = \{I \in D_1 \mid |I| = t\}$. We can write the mass function of the Poisson family as

$$\begin{aligned} q^{Poi}_{(\mathcal{S},\lambda)}(\boldsymbol{\gamma}) &= \textstyle\sum_{\boldsymbol{v}\in\tau^{-1}(\boldsymbol{\gamma})} h_\lambda(\boldsymbol{v}) \\ &= m_{D_0}\left[1 - \textstyle\sum_{t=1}^{|D_0|}(-1)^{t-1}\sum_{I\subseteq\mathcal{I}_t}\exp(-\sum_{k\in\cap_{i\in I}S_i\setminus\cup_{j\in D_1}S_j}\lambda_k)\right]. \end{aligned}$$

For a proof see Appendix 10.

8.3 Fitting the parameter

We need to determine the family of index sets $\mathcal{M}$ and the Poisson parameters $\lambda = (\lambda_1, \dots, \lambda_n)$ such that the resulting family $q_{(\mathcal{S},\lambda)}$ is optimal in terms of distance to the mean and correlation. Obviously, we face a rather difficult combinatorial problem. Park et al. (1996) describe a greedy algorithm, based on convolutions of Poisson variables, that finds at least some feasible combination of $\mathcal{S}$ and $\boldsymbol{\lambda}$.

8.4 Properties

We check the requirement list from Section 1.4:

(a) The Poisson reduction family is not necessarily parsimonious. The number of parameters $\dim(\theta)$ is determined by the fitting algorithm.

(b) We fit the family via method of moments using a fast but non-optimal greedy algorithm.

(c) We sample $\boldsymbol{y} \sim q^{\mathrm{Poi}}_{(\mathcal{S},\lambda)}$ using $\boldsymbol{y} = \tau_{\mathcal{S}}(\boldsymbol{v})$ with $\boldsymbol{v} \sim h_{\boldsymbol{\lambda}}$.

(d) We cannot evaluate $q^{\mathrm{Poi}}_{(\mathcal{S},\lambda)}(\boldsymbol{y})$ since it requires summation of $2^{d-|\boldsymbol{y}|} - 1$ terms using an inclusion-exclusion principle which is computationally not feasible.

(e) The family $q^{\text{Poi}}_{(\mathcal{S},\lambda)}$ can partially reproduce the mean and certain correlation structures of the data $\mathbf{X}$. We cannot sample negative correlations.

Since the family is limited to certain patterns of non-negative correlations, we cannot use it as general-purpose family in adaptive Monte Carlo algorithms. It might be useful, however, if we know that the target distribution π has strictly non-negative correlations.

9 The Archimedean copula family

Genest and Neslehova (2007) discuss in detail the potentials and pitfalls of applying copula theory, which is well developed for bivariate, continuous random variables, to multivariate discrete distribution. Yet, there have been earlier attempts to sample binary vectors via copulas: Lee (1993) describes how to construct an Archimedean copula, more precisely the Frank family, (see e.g. Nelsen (2006, p.119)), for sampling multivariate binary data.

Unfortunately, most results in copula theory do not easily extend to high dimensions. Indeed, we need to solve a non-linear equation for each component when generating a random vector from the Frank copula, and Lee (1993) acknowledges that this is only applicable for $d \leq 3$. For low-dimensional problems, however, we can just enumerate the solution space $\mathbb{B}^d$ and draw from an alias table (Walker, 1977), which somewhat renders the Archimedean copula approach an interesting exercise, but without much practical value in Monte Carlo applications.

References

Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, (72):1–10.

Bahadur, R. (1961). A representation of the joint distribution of responses to n dichotomous items. In Solomon, H., editor, *Studies in Item Analysis and Prediction*, pages pp. 158–68. Stanford University Press.

Bishop, Y., Fienberg, S., and Holland, P. (1975). *Discrete multivariate analysis: Theory and Practice*. Cambridge, MA: MIT Press.

Christensen, R. (1997). *Log-linear models and logistic regression*. Springer Verlag.

Cox, D. and Wermuth, N. (1994). A note on the quadratic exponential binary distribution. *Biometrika*, 81(2):403–408.

Cox, D. and Wermuth, N. (2002). On some models for multivariate binary variables parallel in complexity with the multivariate Gaussian distribution. *Biometrika*, 89(2):462.

Divgi (1979). Computation of univariate and bivariate normal probability functions. *The Annals of Statistics*, (7):903–910.

Drezner, Z. and Wesolowsky, G. O. (1990). On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, (35):101–107.

Emrich, L. and Piedmonte, M. (1991). A method for generating high-dimensional multivariate binary variates. *The American Statistician*, 45:302–304.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, (80):27–38.

Gange, S. (1995). Generating Multivariate Categorical Variates Using the Iterative Proportional Fitting Algorithm. *The American Statistician*, 49(2).

Genest, C. and Neslehova, J. (2007). A primer on copulas for count data. *Astin Bulletin*, 37(2):475.

Haberman, S. (1972). Algorithm AS 51: Log-linear fit for contingency tables. *Applied Statistics*, pages 218–225.

Higham, N. J. (2002). Computing the nearest correlation matrix — a problem from finance. *IMA Journal of Numerical Analysis*, (22):329–343.

Joe, H. (1996). Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters. *Lecture Notes-Monograph Series*, 28:120–141.

Johnson, N., Kotz, S., and Balakrishnan, N. (2002). *Continuous multivariate distributions - models and applications*, volume 2. New York: John Wiley & Sons,.

Lee, A. (1993). Generating Random Binary Deviates Having Fixed Marginal Distributions and Specified Degrees of Association. *The American Statistician*, 47(3).

Leisch, F., Weingessel, A., and Hornik, K. (1998). On the generation of correlated artificial binary data. Technical report, WU Vienna University of Economics and Business.

Mikosch, T. (2006). Copulas: Tales and facts. *Extremes*, 9(1):3–20.

Nelsen, R. (2006). *An introduction to copulas*. Springer Verlag.

Park, C., Park, T., and Shin, D. (1996). A simple method for generating correlated binary variates. *The American Statistician*, 50(4).

Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events. *European Journal of Operations Research*, 99:89–112.

Streitberg, B. (1990). Lancaster interactions revisited. *The Annals of Statistics*, 18(4):1878–1885.

Walker, A. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 3(3):256.

10 Appendix

Proof Proposition 4.1. Recall that $\mathcal{I} = 2^D$ and $v_I(\boldsymbol{\gamma}) = \prod_{i\in I}[(\gamma_i - m_i)/\sqrt{m_i(1-m_i)}]$ with $m_i > 0$ for all $i \in D$. We define an inner product

$$(f,g) := \mathbb{E}_{q_{\boldsymbol{m}}^{\mathrm{Prod}}}\left(f(\boldsymbol{\gamma})g(\boldsymbol{\gamma})\right) = \textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} f(\boldsymbol{\gamma})g(\boldsymbol{\gamma})\prod_{i\in D} m_i^{\gamma_i}(1-m_i)^{1-\gamma_i}$$

on the vector space of real-valued functions on $\mathbb{B}^d$. The set $S = \{v_I(\boldsymbol{\gamma}) \mid I \in \mathcal{I}\}$ is orthonormal, since

$$\begin{aligned}(v_I, r_J) &= \prod_{i\in I\cap J}\mathbb{E}_{q_{\boldsymbol{m}}^{\mathrm{Prod}}}\left(\frac{(\gamma_i-m_i)^2}{m_i(1-m_i)}\right)\prod_{i\in(I\cup J)\setminus(I\cap J)}\mathbb{E}_{q_{\boldsymbol{m}}^{\mathrm{Prod}}}\left(\frac{\gamma_i-m_i}{\sqrt{m_i(1-m_i)}}\right)\\ &= \begin{cases}0 & \text{for } I\neq J\\ 1 & \text{for } I = J,\end{cases}\end{aligned}$$

There are $2^d - 1$ elements in S and $q_{\boldsymbol{m}}^{\mathrm{Prod}}(\boldsymbol{\gamma}) > 0$ which implies that $S\cup\{1\}$ is an orthonormal basis of the real-valued function on $\mathbb{B}^d$. It follows that each function $f\colon \mathbb{B}^d \to \mathbb{R}$ has exactly one representation as linear combination of functions in $S\cup\{1\}$ which is $f = (f,1) + \sum_{I\in\mathcal{I}} v_I(f, v_I)$. Since

$$(\pi/q_{\boldsymbol{m}}^{\mathrm{Prod}}, v_I) = \textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}(\pi(\boldsymbol{\gamma})/q_{\boldsymbol{m})(\boldsymbol{\gamma})}^{\mathrm{Prod}} v_I(\boldsymbol{\gamma}) q_{\boldsymbol{m}}^{\mathrm{Prod}}(\boldsymbol{\gamma}) = \mathbb{E}_\pi\left(v_I(\boldsymbol{\gamma})\right) = c_I,$$

we obtain $\pi(\boldsymbol{\gamma})/q_{\boldsymbol{m}}^{\mathrm{Prod}}(\boldsymbol{\gamma}) = 1 + \sum_{I\in\mathcal{I}} v_I(\boldsymbol{\gamma})\ c_I$ for $f = \pi/q_{\boldsymbol{m}}^{\mathrm{Prod}}$ which concludes the proof. □

Proof Proposition 4.2. We first derive two auxiliary results to structure the proof.

Lemma 1. For a set $I \subseteq D$ of indices it holds that

$$\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}\prod_{k\in I\cup\{i,j\}}\gamma_k = 2^{d-|I|-2+\mathbb{1}_I(i)+\mathbb{1}_{I\cup\{i\}}(j)}.$$

For an index set $M \subseteq D$, we have the sum formula $\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}\prod_{k\in M}\gamma_k = 2^{d-|M|}$. If we have an empty set $M = \emptyset$ the sum equals 2^d and each time we add a new index $i \in D\setminus M$ to M half of the addends vanish. The number of elements in $M = I\cup\{i,j\}$ is the number of elements in I plus one if $i \notin I$ and again plus one if $i \neq j$ and $j \notin I$. Written using indicator function, we have $|I\cup\{i,j\}| = |I| + \mathbb{1}_{D\setminus I}(i) + \mathbb{1}_{D\setminus(I\cup\{i\})}(j) = |I| + 2 - \mathbb{1}_I(i) - \mathbb{1}_{I\cup\{i\}}(j)$ which implies Lemma 1. □

Lemma 2.

$$\sum_{i\in D}\sum_{j\in D}2^{\mathbb{1}_I(i)+\mathbb{1}_{I\cup\{i\}}(j)}\ a_{i,j} = \mathbf{1}^\intercal\mathbf{A}\mathbf{1} + \operatorname{tr}[\mathbf{A}] + \sum_{i\in I}\left[2\sum_{j\in D}a_{i,j} + \sum_{j\in I\setminus\{i\}}a_{i,j}\right]$$

Straightforward calculations:

$$\begin{aligned}2^{\mathbb{1}_I(i)+\mathbb{1}_{I\cup\{i\}}(j)} &= (1+\mathbb{1}_I(i))(1+\mathbb{1}_{I\cup\{i\}}(j))\\ &= (1+\mathbb{1}_I(i))(1+\mathbb{1}_I(j)+\mathbb{1}_{\{i\}}(j)-\mathbb{1}_{I\cap\{i\}}(j))\\ &= 1+\mathbb{1}_I(i)+\mathbb{1}_I(j)+\mathbb{1}_I(i)\mathbb{1}_I(j)\\ &\qquad+\mathbb{1}_{\{i\}}(j)+\mathbb{1}_I(i)\mathbb{1}_{\{i\}}(j)-\mathbb{1}_{I\cap\{i\}}(j)-\mathbb{1}_I(i)\mathbb{1}_{I\cap\{i\}}(j)\\ &= 1+\mathbb{1}_{\{i\}}(j)+\mathbb{1}_I(i)+\mathbb{1}_I(j)+\mathbb{1}_{I\times I}(i,j)-\mathbb{1}_{I\cap\{i\}}(j),\end{aligned}$$

where we used the identity

$$\mathbb{1}_I(i)\mathbb{1}_{\{i\}}(j) = \mathbb{1}_I(i)\mathbb{1}_I(i)\mathbb{1}_{\{i\}}(j) = \mathbb{1}_I(i)\mathbb{1}_I(j)\mathbb{1}_{\{i\}}(j) = \mathbb{1}_I(i)\mathbb{1}_{I\cap\{i\}}(j)$$

in the second line. Thus, we have

$$\begin{aligned}
&\sum_{i\in D}\sum_{j\in D} 2^{\mathbb{1}_I(i)+\mathbb{1}_{I\cup\{i\}}(j)}\ a_{i,j} \\
&= \sum_{i\in D}\sum_{j\in D} \left(1 + \mathbb{1}_{\{i\}}(j) + \mathbb{1}_I(i) + \mathbb{1}_I(j) + \mathbb{1}_{I\times I}(i,j) - \mathbb{1}_{I\cap\{i\}}(j)\right)\ a_{i,j} \\
&= \sum_{i\in D}\sum_{j\in D} a_{i,j} + \sum_{j\in D} a_{j,j} + \sum_{i\in I}\sum_{j\in D} a_{i,j} + \sum_{i\in D}\sum_{j\in I} a_{i,j} + \sum_{i\in I}\sum_{j\in I} a_{i,j} - \sum_{i\in I} a_{j,j} \\
&= \mathbf{1}^\intercal\mathbf{A}\mathbf{1} + \mathrm{tr}\,[\mathbf{A}] + \sum_{k\in I}\left[2\sum_{l\in D} a_{k,l} + \sum_{l\in I} a_{k,l} - a_{k,k}\right] \\
&= \mathbf{1}^\intercal\mathbf{A}\mathbf{1} + \mathrm{tr}\,[\mathbf{A}] + \sum_{k\in I}\left[2\sum_{l\in D} a_{k,l} + \sum_{l\in I\setminus\{k\}} a_{k,l}\right]
\end{aligned}$$

The last line is the assertion of Lemma 2. □

Using the two Lemmata above, we find a convenient expression for the cross-moment

$$\begin{aligned}
m_I &= \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} (\prod_{k\in I}\gamma_k)\ \mu(a_0 + \boldsymbol{\gamma}^\intercal\mathbf{A}\boldsymbol{\gamma}) \\
&= \mu\left[\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} a_0 + \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}(\prod_{k\in I}\gamma_k)\sum_{i\in D}\sum_{j\in D}\gamma_i\gamma_j\ a_{i,j}\right] \\
&= \mu\left[2^{d-|I|}a_0 + \sum_{i\in D}\sum_{j\in D} a_{i,j}\ \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}(\prod_{k\in I\cup\{i,j\}}\gamma_k)\right] \text{(Lemma 1)} \\
&= \mu\left[2^{d-|I|}a_0 + \sum_{i\in D}\sum_{j\in D} 2^{d-|I\cup\{i,j\}|}\ a_{i,j}\right] \\
&= \mu 2^{d-|I|-2}\left[4a_0 + \sum_{i\in D}\sum_{j\in D} 2^{\mathbb{1}_I(i)+\mathbb{1}_{I\cup\{i\}}(j)}\ a_{i,j}\right] \text{(Lemma 2)} \\
&= \mu 2^{d-|I|-2}\left[4a_0 + \mathbf{1}^\intercal\mathbf{A}\mathbf{1} + \mathrm{tr}\,[\mathbf{A}] + \sum_{i\in I}\left[2\sum_{j\in D} a_{i,j} + \sum_{j\in I\setminus\{i\}} a_{i,j}\right]\right]
\end{aligned}$$

Since $m_\emptyset = 1$ by definition, we the normalizing constant is

$$\mu = 2^{-d+2}\left(4a_0 + \mathbf{1}^\intercal\mathbf{A}\mathbf{1} + \mathrm{tr}\,[\mathbf{A}]\right)^{-1},$$

which allows us to write down the normalized cross-moments

$$m_I = \frac{1}{2^{|I|}} + \frac{\sum_{i\in I}\left[2\sum_{j\in D} a_{i,j} + \sum_{j\in I\setminus\{i\}} a_{i,j}\right]}{2^{|I|}(4a_0 + \mathbf{1}^\intercal\mathbf{A}\mathbf{1} + \mathrm{tr}\,[\mathbf{A}])}.$$

The proof is complete. □

Proof Proposition 4.4. We margin out the last component d. Let $I = \{1, \dots, d-t\}$,

$$\begin{aligned}
q^{(d-1)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_I)\mu^{-1} &= \left(q^{(d)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_I, 1) + q^{(d)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_I, 0)\right)\mu^{-1} \\
&= 2a_0 + (\boldsymbol{\gamma}_I, 1)^\intercal \mathbf{A}(\boldsymbol{\gamma}_I, 1) + (\boldsymbol{\gamma}_I, 0)^\intercal \mathbf{A}(\boldsymbol{\gamma}_I, 0) \\
&= 2a_0 + \mathrm{tr}\left[\mathbf{A}\left[(\boldsymbol{\gamma}_I, 1)(\boldsymbol{\gamma}_I, 1)^\intercal + (\boldsymbol{\gamma}_I, 0)(\boldsymbol{\gamma}_I, 0)^\intercal\right]\right] \\
&= 2a_0 + \mathrm{tr}\left[\mathbf{A}\begin{bmatrix} 2\boldsymbol{\gamma}_I\boldsymbol{\gamma}_I^\intercal & \boldsymbol{\gamma}_I \\ \boldsymbol{\gamma}_I^\intercal & 1 \end{bmatrix}\right]
\end{aligned}$$

Iterating the argument, we obtain for $I = \{1, \dots, d-t\}$ and $I^c = D \setminus I$

$$q^{(d-t)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_I)\mu^{-1} = 2^t a_0 + 2^{t-2}\,\mathrm{tr}\left[\mathbf{A}\begin{bmatrix} 4\,\boldsymbol{\gamma}_I\boldsymbol{\gamma}_I^\intercal & 2\,\boldsymbol{\gamma}_I\mathbf{1}_t^\intercal \\ 2\,\mathbf{1}_t\boldsymbol{\gamma}_I^\intercal & \mathbf{1}_t\mathbf{1}_t^\intercal + \mathbf{I}_t \end{bmatrix}\right]$$

Straightforward calculations:

$$\begin{aligned}
&\mathrm{tr}\left[\mathbf{A}\begin{bmatrix} 4\,\boldsymbol{\gamma}_I\boldsymbol{\gamma}_I^\intercal & 2\,\boldsymbol{\gamma}_I\mathbf{1}_t^\intercal \\ 2\,\mathbf{1}_t\boldsymbol{\gamma}_I^\intercal & \mathbf{1}_t\mathbf{1}_t^\intercal + \mathbf{I}_t \end{bmatrix}\right] \\
&= \mathrm{tr}\left[\mathbf{A}\left[(2\,\boldsymbol{\gamma}_I, \mathbf{1}_t)(2\,\boldsymbol{\gamma}_I, \mathbf{1}_t)^\intercal + \mathrm{diag}\left[\mathbf{0}_I, \mathbf{1}_t\right]\right]\right] \\
&= \left[(2\,\boldsymbol{\gamma}_I, \mathbf{1}_t)^\intercal \mathbf{A}(2\,\boldsymbol{\gamma}_I, \mathbf{1}_t) + \mathrm{tr}\left[\mathbf{A}\mathrm{diag}\left[\mathbf{0}_I, \mathbf{1}_t\right)\right]\right] \\
&= \left[4\sum_{i\in I}\sum_{j\in I}\gamma_i\gamma_j a_{i,j} + 4\sum_{i\in I}\sum_{j\in I^c}\gamma_i a_{i,j} + \sum_{i\in I^c}\sum_{j\in I^c} a_{i,j} + \sum_{i\in I^c} a_{i,i}\right] \\
&= \left[4\sum_{i\in I}\gamma_i\Big(\sum_{j\in I}\gamma_j a_{i,j} + \sum_{j\in I^c} a_{i,j}\Big) + \sum_{i\in I^c}\sum_{j\in I^c} a_{i,j} + \sum_{i\in I^c} a_{i,i}\right]
\end{aligned}$$

The proof is complete. ∎

Proof Proposition 5.1. For convenience of notation, let $\boldsymbol{\gamma}_- = (\gamma_1, \dots, \gamma_{d-1})$. Note that $q_{\mathbf{A}}(\boldsymbol{\gamma}) = \mu\exp(\boldsymbol{\gamma}_-^\intercal\mathbf{A}'\boldsymbol{\gamma}_- + \gamma_d(2\boldsymbol{b}^\intercal\boldsymbol{\gamma}_- + c))$. The marginal distribution is therefore

$$\begin{aligned}
\pi(\boldsymbol{\gamma}_-) &= \mu\,\exp(\boldsymbol{\gamma}_-^\intercal\mathbf{A}'\boldsymbol{\gamma}_-)\left(1 + \exp(2\boldsymbol{\gamma}_-^\intercal\boldsymbol{b} + c)\right) \\
&= \mu\,\exp\left(\boldsymbol{\gamma}_-^\intercal\mathbf{A}'\boldsymbol{\gamma}_- + \boldsymbol{\gamma}_-^\intercal\boldsymbol{b} + \frac{c}{2}\right)\left(\exp(-\boldsymbol{\gamma}_-^\intercal\boldsymbol{b} - \frac{c}{2}) + \exp(\boldsymbol{\gamma}_-^\intercal\boldsymbol{b} + \frac{c}{2})\right) \\
&= \mu\,\exp\left(\boldsymbol{\gamma}_-^\intercal\mathbf{A}'\boldsymbol{\gamma}_- + \boldsymbol{\gamma}_-^\intercal\boldsymbol{b} + \frac{c}{2}\right)2\cosh\left(\boldsymbol{\gamma}_-^\intercal\boldsymbol{b} + \frac{c}{2}\right).
\end{aligned}$$

The marginal log mass function is thus

$$\log\pi(\boldsymbol{\gamma}_-) = \log(2\mu) + \frac{c}{2} + \boldsymbol{\gamma}_-^\intercal\mathbf{A}'\boldsymbol{\gamma}_- + \boldsymbol{\gamma}_-^\intercal\boldsymbol{b} + \log\cosh\left(\boldsymbol{\gamma}_-^\intercal\boldsymbol{b} + \frac{c}{2}\right).$$

For $\log\cosh$ we can use a Taylor approximation

$$\log\cosh(\boldsymbol{\gamma}_-^\intercal\boldsymbol{b} + \frac{c}{2}) \approx \log\cosh(\frac{c}{2}) + \boldsymbol{\gamma}_-^\intercal\boldsymbol{b}\,\tanh(\frac{c}{2}) + \frac{1}{2}\,(\boldsymbol{\gamma}_-^\intercal\boldsymbol{b})^2\,\mathrm{sech}^2(\frac{c}{2})$$

to obtain

$$\begin{aligned}
\log\pi(\boldsymbol{\gamma}_-) \approx{}& \log(2\mu\cosh(\frac{c}{2})) + \frac{c}{2} + \boldsymbol{\gamma}_-^\intercal\mathbf{A}'\boldsymbol{\gamma}_- \\
&+ \left(1 + \tanh(\frac{c}{2})\right)\boldsymbol{\gamma}_-^\intercal\boldsymbol{b} + \frac{1}{2}\,\mathrm{sech}^2(\frac{c}{2})(\boldsymbol{\gamma}_-^\intercal\boldsymbol{b})^2
\end{aligned}$$

Since $\boldsymbol{\gamma}_-$ is a binary vector, we have $\boldsymbol{\gamma}_-^\intercal \boldsymbol{b} = \boldsymbol{\gamma}_-^\intercal \mathrm{diag}\,[\boldsymbol{b}]\,\boldsymbol{\gamma}_-$ and can thus rewrite the inner products as

$$\begin{aligned}\boldsymbol{\gamma}_-^\intercal \mathbf{A}'\boldsymbol{\gamma}_- + \gamma_-^\intercal \boldsymbol{b} + (\gamma_-^\intercal \boldsymbol{b})^2 &= \mathrm{tr}\left[\mathbf{A}'\boldsymbol{\gamma}_-\boldsymbol{\gamma}_-^\intercal + \mathrm{diag}\,[\boldsymbol{b}]\,\boldsymbol{\gamma}_-\boldsymbol{\gamma}_-^\intercal + \boldsymbol{b}\boldsymbol{b}^\intercal\boldsymbol{\gamma}_-\boldsymbol{\gamma}_-^\intercal\right] \\ &= \boldsymbol{\gamma}_-^\intercal(\mathbf{A}' + \mathrm{diag}\,[\boldsymbol{b}] + \boldsymbol{b}\boldsymbol{b}^\intercal)\boldsymbol{\gamma}_-.\end{aligned}$$

We let denote

$$\mu^* = 2\mu\cosh(\frac{c}{2})\exp(\frac{c}{2}) = \mu(\exp(-\frac{c}{2}) + \exp(\frac{c}{2}))\exp(\frac{c}{2}) = \mu\,(1+\exp(c))$$

and

$$\mathbf{A}^* = \mathbf{A}' + \Big(1+\tanh(\frac{c}{2})\Big)\,\mathrm{diag}\,[\boldsymbol{b}] + \frac{1}{2}\,\mathrm{sech}^2(\frac{c}{2})\boldsymbol{b}\boldsymbol{b}^\intercal$$

to form the approximation $\pi(\boldsymbol{\gamma}_-) \approx \mu^*\,\exp(\boldsymbol{\gamma}_-^\intercal\mathbf{A}^*\boldsymbol{\gamma}_-)$ which completes the proof.

□

Proof Proposition 8.1. Straightforward calculations using an inclusion-exclusion argument for the union of events:

$$\begin{aligned} q_{(\mathcal{S},\lambda)}(\boldsymbol{\gamma}) &= \sum_{\boldsymbol{v}\in\tau^{-1}(\boldsymbol{\gamma})} h_\lambda(\boldsymbol{v}) = \mathbb{P}_{h_\lambda}\left(\cap_{i\in D}\left\{\mathbb{1}_{\{0\}}\textstyle\sum_{k\in S_i} v_k = \gamma_i\right\}\right) \\ &= \mathbb{P}_{h_\lambda}\left(\cap_{i\in D_1}\cap_{k\in S_i}\{v_k = 0\}\,,\ \cap_{i\in D_0}\cup_{k\in S_i}\{v_k > 0\}\right) \\ &= \mathbb{P}_{h_\lambda}\left(\cap_{i\in D_1}\cap_{k\in S_i}\{v_k = 0\}\right)\mathbb{P}_{h_\lambda}\Big(\cap_{i\in D_0}\cup_{k\in S_i\setminus\cup_{j\in D_1}S_j}\{v_k > 0\}\Big) \\ &= \mathbb{P}_{q_{(\mathcal{S},\lambda)}}\left(\boldsymbol{\gamma}_{D_1} = \mathbf{1}\right)\Big(1 - \mathbb{P}_{h_\lambda}\Big(\cup_{i\in D_0}\cap_{k\in S_i\setminus\cup_{j\in D_1}S_j}\{v_k = 0\}\Big)\Big) \\ &= m_{D_0}\left[1 - \sum_{t=1}^{|D_0|}(-1)^{t-1}\sum_{I\subseteq\mathcal{I}_t}\mathbb{P}_{h_\lambda}\Big(\cap_{i\in I}\cap_{k\in S_i\setminus\cup_{j\in D_1}S_j}\{v_k = 0\}\Big)\right] \\ &= m_{D_0}\left[1 - \sum_{t=1}^{|D_0|}(-1)^{t-1}\sum_{I\subseteq\mathcal{I}_t}\exp\left(\sum_{k\in\cap_{i\in I}S_i\setminus\cup_{j\in D_1}S_j} -\lambda_k\right)\right].\end{aligned}$$

The proof is complete. □